* Feature Detection & Matching

* Applications of Feature detection & mapping

- Image alignment → ☑ Image mosaicing
- Correspondence → ☑ 3D Models, In-between image

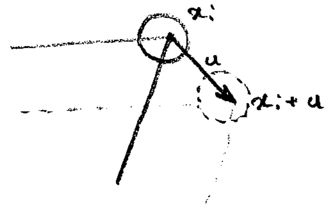
* Types of features

- Interest points (Keypoint features) + Region-based features
  - ↳ The appearance of patches of pixels surronding the point location
    - ☑ Corners, interestingly shaped patches
  - ↳ localized
- Edges
  - ↳ orientation & local appearance (edge profile)
  - ↳ indicators for object boundaries & occlusion
  - ↳ grouped into curves and straight line segments
    - ↳ vanishing point → internal & external camera params

* Point Features

- Finding sparse set of corresponding locations in different images
  - ↳ computing camera pos
  - ↳ stereo matching
  - ↳ align different images
    - → ☑ automated 3D modeling
    - → ☑ stitching image mosaics
    - → ☑ video stabilization
  - ↳ object instance & category recognition
- Advantage
  - ↳ works with clutter /occlusion
  - ↳ works with large scale and orientation changes
- Approaches
  1. Find features in one image → can be accurately tracked
     - ↳ tracking use local search technique
       - → correlation
       - → least squares
     - ↳ suitable
       - → images taken from nearby view points
       - → images taken from rapid succession → video sequence

- patches with gradients in at least two different directions
  - ↳ significantly
  - ↳ easiest to localize



u: displacement

x_i : patch center

$\omega(x_i)$: circle
↳ weighting function
↳ patch window

- weighted summed square difference

$$E_{WSSD}(u) = \underbrace{\sum_i}_{\text{all pixels in the patch}} \underbrace{\omega(x_i)}_{\text{spatially varying weighting function}} \Big[\underbrace{I_1}_{\text{img 1}}(x_i + \underbrace{u}_{\text{displacement}}) - \underbrace{I_0}_{\text{img 0}}(x_i) \Big]^2$$

- how good is this metric?
  - ↳ it can be matched to other patches in image
  - ↳ auto-correlation function or surface → to compute how stable this metric is wit respect to small variations in position, Δu

---

$$E_{AC}(\Delta u) = \sum_i \omega(x_i) \left[I_0(x_i + \Delta u) - I_0(x_i) \right]^2$$

↳ SSD surface : weighted sum of squared diff

↳ correlation of patches → normally = product of two patches (not here!)

↳ E_{AC} → good unique min → good feature
→ several minimum → Aperture problem
→ ...
→ no good peak → bad textureless feature

Plot : 3D
D1 : u_x
D2 : u_y
D3 : E_{AC}

↳ Tylor series expansion $I_0(x_i + \Delta u) \approx I_0(x_i) + \nabla I_0(x_i) \cdot \Delta u$

$$E_{AC}(\Delta u) \approx \sum_i w_i \left[\nabla I_0(x_i) \cdot \Delta u \right]^2 = \vec{\Delta u}^T A \vec{\Delta u}$$

$$\nabla I_0(x_i) = \left(\frac{\partial I_0}{\partial x}, \frac{\partial I_0}{\partial y} \right)(x_i) \longrightarrow \text{image gradient}$$

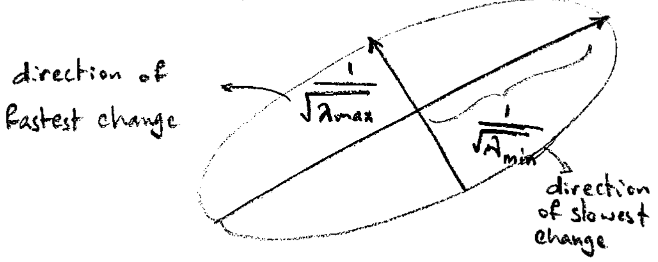
↳ image gradient → Harris detector → [-2 -1 0 +1 +1)
→ convolve with horizontal + vertical derivatives of Gaussian usually $\sigma = 1$

↳ Auto correlation matrix

$$A = w * \begin{bmatrix} I_x^2 & I_xI_y \\ I_xI_y & I_y^2 \end{bmatrix} \longrightarrow \text{tensor image}$$

↳ The inverse of A provides a lower bound on the uncertainty in the location of matching path → useful indicator of which patches can be reliably matched

↳ Eigenvalue analysis of A → (λ_0, λ_1) and eigenvectors



smaller eigenvalue ⟹ larger uncertaity

uncertaity $\propto \left(\frac{1}{\sqrt{\lambda_0}}\right)$

maximum λ_0 ⟹ lowest uncertaity
better feature to track

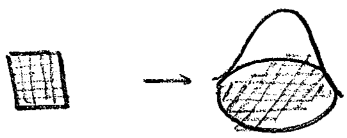
⟹ Simply : Find all patches → form A → calc λ_0

Biggest λ_0 ⟹ best patch

explained in back of the page

— Förstner-Harris → propose same idea in rotationally invariant scalar measures
↳ derived from auto-correlation matrix
↳ for sparse feature matching

↳ instead of square patch → we can use gaussian patch



makes detector insensitive to in-plane image rotations

— Harris-Stephans : find keypoint features use

$$\det(A) - \alpha\, \mathrm{trace}(A)^2 \qquad (\alpha = 0.06)$$

↳ rotationally invariant

↳ does not require the use of $\sqrt{}$ → $\det(A) - \alpha\, tr(A)^2 = \lambda_0\lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$

↳ down weights edge-like features when $\lambda_1 \gg \lambda_0$

— Triggs : find keypoint features using

$$\lambda_0 - \alpha\lambda_1 \qquad (\alpha = 0.05)$$

↳ reduces the response at 1D edges → prevent aliasing error that makes small eigen value bigger

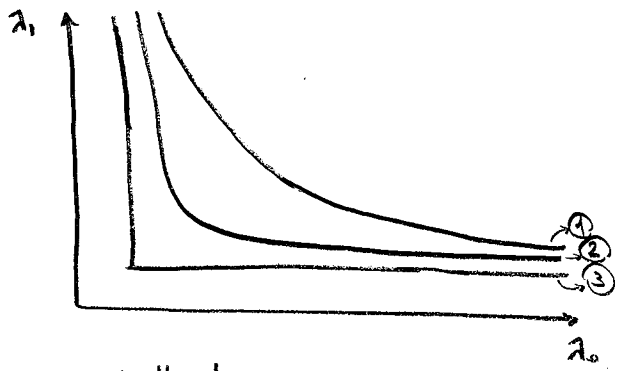
- Brown et. al: keypoint feature detection

$$\frac{\det(A)}{\operatorname{tr}(A)} = \frac{\lambda_0 \lambda_1}{\lambda_0 + \lambda_1}$$

↳ harmonic mean
↳ smoother where $\lambda_0 \approx \lambda_1$ (circle → no correlation)

- Compare popular keypoint feature detectors

1. Harris → $\lambda_0 \lambda_1 - \alpha(\lambda_0 + \lambda_1)^2$
2. Harmonic mean → $\frac{\lambda_0 \lambda_1}{\lambda_0 + \lambda_1}$
3. Shi_Tomasi → $\frac{1}{\sqrt{\lambda_0}}$



↳ all of them loos for points where λ_0 and λ_1 are both large

- Basic feature detection Alg

1. Compute I_x & I_y → derivatives of image in x & y axis
   ↘ = convoluting image with derivatives of Gaussian

2. compute three images corresponding to the outer product of thess gradients → $I_x \cdot I_x$, $I_y \cdot I_y$, $I_x \cdot I_y$

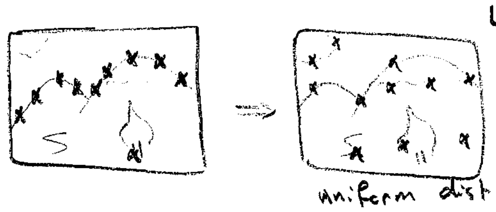
   ↳ $A = \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$

3. Convolve each of these images with a larger Gaussian

4. Compute a scalar interest measure → using formulas above

5. Find local maxima above a certain threshold → Report them as detected feature point locat
   - local maxima ↳ not all of the points
   - certain threshold ↳ only high prob. candidates

- ANMS : Active Non-Maximal Supression
  ↳ Most feature detector → find maxima → uneven feature dist. across image
  ↳ Mitigation → local maxima & response value greater than neighborhood
     ↳ neighborhood : radius r
     ↳ greaters : 10% more significant (at least)
     ↳ efficient implementation: sort local maxima by response strengh
        then sort by decreasing suppresion radius



uniform dist

2. Independently detect features in all images → then match them

↳ matching based on local appearance

↳ suitable → large amount of motion / appearance change expected

☑ stitching together panoramas

correspondance in wide baseline stereo

object recognition

- Stages of keypoint detection

1. Feature detection / extraction

↳ search image for locations

↳ locations: locations likely to match well in other images

2. Feature description

↳ description of region around detected key point

↳ convert it to compact and stable / invariant descriptor

↳ descriptors can be matched together

3. Feature Matching

↳ searching for likely matching candidates in other images

↳ efficient

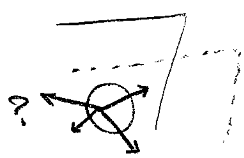
3'. Feature Tracking

↳ like feature matching → but searches small neighborhood

↳ suitable for video processing

☑ SIFT is good example

- what is good feature?



– textureless patches → impossible to match / track

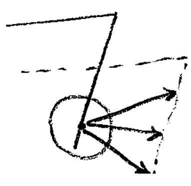
– large contrast change in patch → gradient

↳ easier to localize

↳ aperture problem: straight line segments at a single orientation

↳ only possible to align patches along the direction normal to the edge

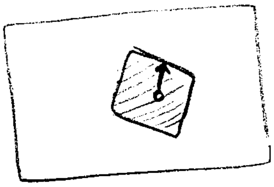
barber-pole illusion ←



- Measuring Repeatability
  - repeatability : frequency with which keypoint detected in one image are found within $\mathcal{E}$ pixels of the corresponding location in transformed image. ($\mathcal{E} = 1.5$)
    - ↳ rotation
      scale change
      illumination change
      viewpoint change
      adding noise
  - information content of each keypoint : entropy of a set of rotationaly invariant local grayscale discriptors.

  ☑ Improved version of Harris detector is good.
  ↳ Gaussian derivative → $\sigma_d = 1$ scale of derivative Gaussian
  → $\sigma_i = 2$ scale of integration Gaussian

- Scale Invariance
  - Detecting features in finest stable scale is not always appropriate
  ☑ matching images that don't have much high-freq details → clouds
  - Solution 1 : Extract features at variaty of scales
    ↳ multiple resolution in pyramid → matching feature at same level
    ↳ suitable : when image don't change scale too much
      ☑ matching successive aerial images taken from airplane
      stitching panorama with fixed-focal-lenght camera
    ↳ MOPS : Multiscale Oriented patches → image orientation
      region of sampling scale

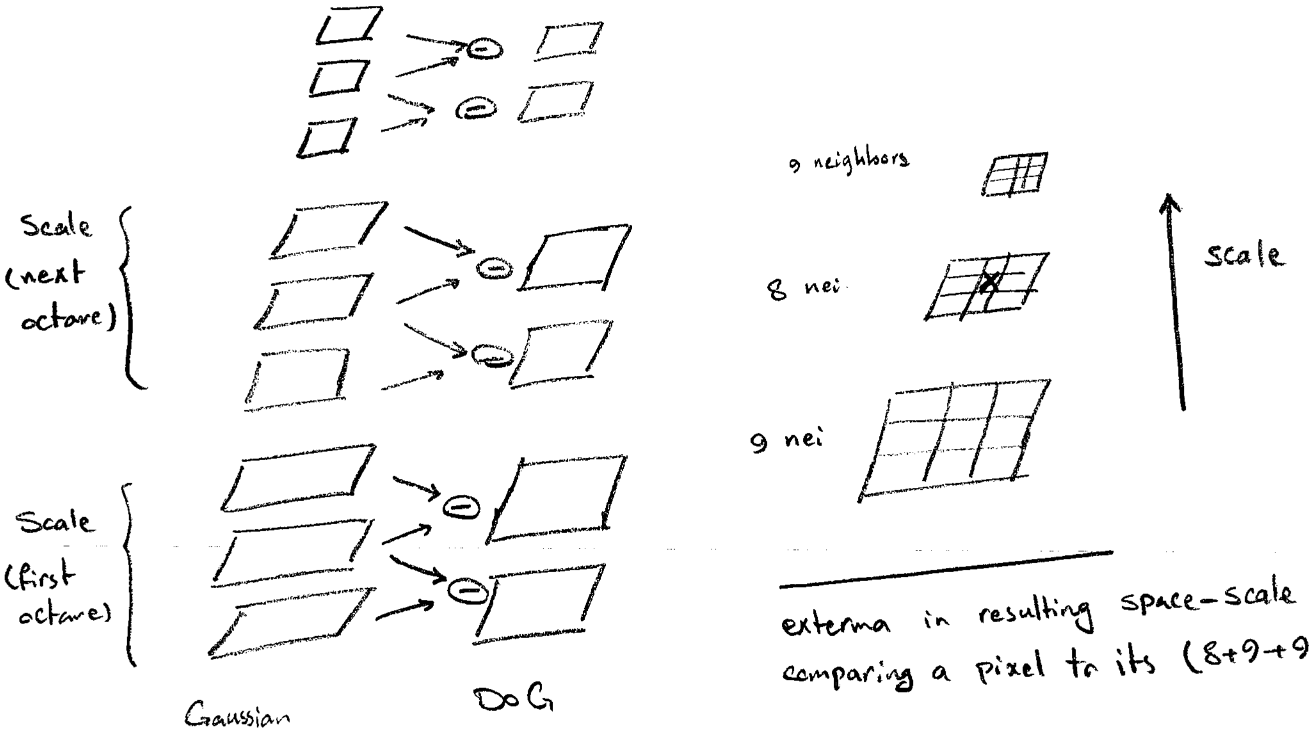


  - Solution 2 : Extract features that are stable in both location & scale
    ↳ suitable : object recognition → scale of object is unknown
    ↳ Lindeberg : Using exterma in Laplacian of Gaussian (LoG) function
    ↳ Lowe : Computing set of sub-octave Difference of Gaussian filters
      looking for space - scale maxima → compute sub-pixel using quadratic fit
        ↳ number of sub-octave levels = 3

    ↳ Eliminating pixels that has strong asymmetry in Harris operator
      ↳ Assymetry in local curvature of DoG function
      ↳ Hessian of Difference image D : $H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}$
        ↳ DoG
      ↳ rejecting keypoints that $\frac{Tr(H)^2}{Det(H)} > 10$

$$a = \frac{Tr(H)^2}{Det(H)} = \frac{D_{xx}^2 + D_{yy}^2}{D_{xx}D_{yy} - D_{xy}^2} \Rightarrow a\uparrow$$

Assymmetry $\longrightarrow$ $D_{xy} \uparrow$



externa in resulting space-scale are detected by comparing a pixel to its (8+9+9 =) 26 neighbors

- SIFT : Scale Invariant Feature Transform
  - ↳ do not use auto-correlation base $\longrightarrow$ can be used to help auto-correlation featur
- Mikolajczyk: add scale selection mechanism to Harris corner detector
  - ↳ evaluating LoG at each detected Harris point
    - ↳ in a multi-scale pyramid
      - ↳ keeps only points which Laplacian is external

  } same idea as figure above

• Rotational Invariance & Orientation Estimation
- Solution 1: design rotationally invariant descriptors
  - ↳ have poor discriminability $\longrightarrow$ map different looking patches to the same descriptor
- Solution 2: dominant orientation
  - ↳ estimate local orientation & scale of keypoint $\longrightarrow$ extract a scaled and oriented patch around keypoint $\longrightarrow$ use it as feature descriptor
  - ↳ Idea 1: average gradient in region

    use guassian weighting func $\longrightarrow$ guassian neighborhood of gradient of point $\simeq$ gradient of point neighborhood smoothed by guassian

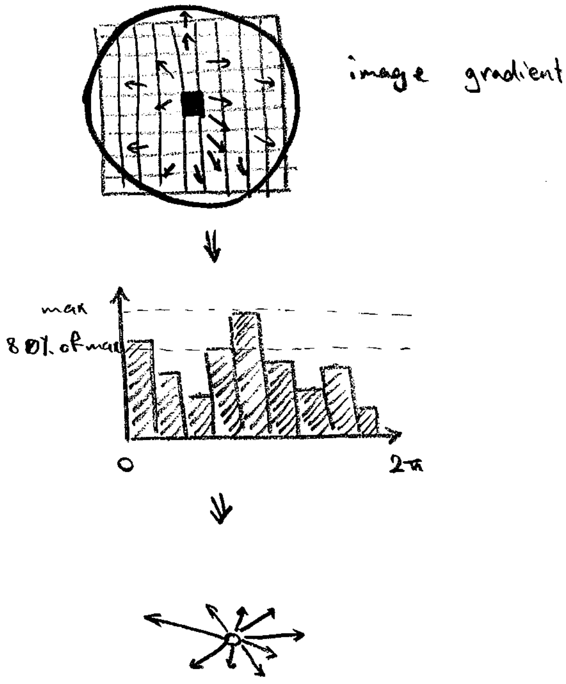
    ↓

    convolve with guassian first, then take gradient

reliable $\Leftrightarrow$ Gaussian kernel size $>$ detection window

↳ Idea 2: Histogram of oriented gradiants (HoG)

1. 36 bins → 10 degree in each bin
2. histogram of weighted edge orientation
   - ↳ gradient magnitude
   - ↳ guassian distance to center
3. find all peaks within 80% of global max
4. compute orientation estimate using a 3-bin parabolic fit



- Affine Invariance
  - Applications: wide baseline stereo matching
    location recognition
  - Affine invariat detector → respond to consistent → scale & orientation change
    ↳ affine deformations
    ☑ perspective foreshortening
  - Solution 1: auto-correlation
    - ↳ calculate auto-correlation or Hessian Matrix
      - ↳ Do eigenvector analysis → fitting an ellipse into the matrix
        - ↳ use principle axes and ratios of this fit as affine coordinate frame

* Invariance

- Usually mixed with covariance and equivariance
- A function

$$f : X \longmapsto Y \qquad g \in G$$

element ↙ ↘ transformation group

- transformation

$$\tau_g^X \quad \text{and} \quad \tau_g^Y$$

☑ rotation → τ_g^X

parameter set → g

- covariance

$$\text{cov} : \forall g \in G : f(\tau_g^X(p)) = \tau_g^Y f(p)$$

$$p \in X$$

☑

$$\tau_g^X = \begin{bmatrix} \cos\alpha & \sin\alpha \\ -\sin\alpha & \cos\alpha \end{bmatrix} \sim\!\!\to g \text{ governs } \alpha$$

meaning : $f(\text{transform}(x)) = \text{transform}' \, f(x)$

- invariance:

$$\text{inv} : \forall g \in G : f(\tau_g^X(p)) = I^Y f(p) = f(p)$$

Input Space X

Output Space Y

$$\begin{tikzcd} \boxed{\text{Pattern } p} \arrow[r, "\tau_g^X"] \arrow[d, "f"] & \boxed{\text{transformed pattern } p' = \tau_g^X p} \arrow[d, "f"] \\ \boxed{f(p)} \arrow[r, "\tau_g^Y"] & \boxed{f(\tau_g^X p) = \tau_g^Y f(p)} \end{tikzcd}$$

$\Rightarrow$ invariance

$$\begin{tikzcd} \boxed{p} \arrow[r, "\tau_g^X"] \arrow[d, "f"] & \boxed{p'} \arrow[dl, "f"] \\ \boxed{f(p)} & \end{tikzcd}$$

- image $\quad \mathbb{I} \in L^2(\mathbb{R}^2) \qquad \mathbb{I}$ is the pattern and $L^2(\mathbb{R}^2)$ is the input space.
- translation

$$t \in \mathbb{R}^2 \qquad (\tau_t \mathbb{I})(r) := \mathbb{I}(r - t)$$

$$\tau_t \in T(2)$$

• rotation

$$(\mathcal{T}_\theta I)(r) := I(u_\theta^T r)$$

$$u_\theta = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$$

$$SO(2) \iff u^T u = I \quad \& \quad \det(u) = 1$$

↳ special orthogonal

$$\mathcal{T}_t \mathcal{T}_\theta I = \mathcal{T}_\theta \mathcal{T}(u_\theta^T t) I$$

• vector valued image

$$f: \mathbb{R}^3 \longmapsto \mathbb{C}^m$$

$$(\mathcal{T}_\theta^X f)(r) := \underbrace{\mathcal{T}_\theta^{\mathbb{C}^m}}_{\text{value transformation}} \underbrace{f(u_\theta^T r)}_{\text{coordinate transformation}}$$

☑ RGB image

$$f: \mathbb{R}^3 \longmapsto \mathbb{R}^3$$

$$(\mathcal{T}_\theta^X f)(r) := I_{3\times3} f(u_\theta^T r)$$

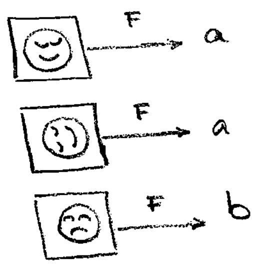
$$\text{if } f := \underbrace{\nabla I}_{\text{image gradient}} \rightarrow (\mathcal{T}_\theta^X f)(r) := u_\theta f(u_\theta^T r)$$

↳ in RGB the channels do not mix, but in the gradient image they do.

• Goal: to have a function that map image to features which are rotation invariant

$$F: L^2(\mathbb{R}^3) \longmapsto \mathbb{C}$$

$$F(\mathcal{T}_\theta I) = F(I)$$

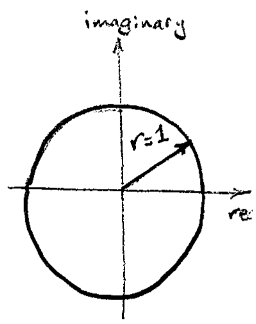


• Rotation group SO(2)

$$R(\varphi) := \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}$$

Isomorph $U(1) := \{ z \in \mathbb{C} \mid |z| = 1 \}$

$$U(1) := \{ e^{i\varphi} \mid \varphi \in [0, 2\pi] \}$$



$$R(\varphi)\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi \\ \sin\varphi & \cos\varphi \end{bmatrix}\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\varphi . X - \sin\varphi . Y \\ \sin\varphi . X + \cos\varphi . Y \end{bmatrix}$$

→ to change the representation to complex numbers (only works in 2D)

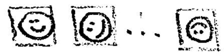
$$e^{i\varphi}(X+iY) = (\cos\varphi + i\sin\varphi)(X+iY) = \cos\varphi\, X - \sin\varphi\, Y + i(\cos\varphi\, Y + \sin\varphi\, X)$$

- Irreducible representation of SO(2)
  - by Haar

$f \in L^2(g)$

$\forall h \in g$

$$\int_g f(hg)\, dg = \int_g f(g) . dg < \infty$$

$\boxed{\checkmark}$ $\int$  $= a < \infty$

- if you rotate your image in all possible directions, and integrate them, the result will be a number (not infinity)
- in the case of rotation the result will be a compact group (= Lie group, ...)
  - ↳ scale, translation is not compact
- for compact group, there will be irreducible representation
  - ↳ like SO(2)

$$\underbrace{Y^{\ell}(\varphi)}_{\text{irreducible representation}} := \underbrace{e^{i\ell\varphi}}_{\text{looks like Fourier series}}, \quad \ell \in \mathbb{N}$$

→ there will be $\underbrace{\text{orthogonal}}_{OB}$ basis for that

$$\{Y^{\ell}\}_{\ell=0,1,\ldots} \text{ OB for } \underbrace{f \in L^2(SO(2))}_{f(\varphi) \mapsto \mathbb{C}}$$

$$\langle Y^{\ell_1}, Y^{\ell_2} \rangle = 2\pi\, \delta_{\ell_1,\ell_2}$$

$$\delta_{\ell_1,\ell_2} : \begin{cases} 1 & , \ell_1 = \ell_2 \\ 0 & , \ell_1 \neq \ell_2 \end{cases}$$

} orthogonality

- invariant subspace: if T_g is acting on V, if V' is a subspace of that which is invariant to T_g

$$V' \subseteq V \Rightarrow \forall v' \in V' : (T_g v') \in V'$$ → the subspace is closed to T_g

- fourier analogy

$$f(\varphi) = \frac{1}{\sqrt{2\pi}} \sum_{\ell} a^{\ell} Y^{\ell}(\varphi) \quad , a^{\ell} \in \mathbb{C}$$

$$a^{\ell} = \langle f, \frac{1}{\sqrt{2\pi}} Y^{\ell} \rangle$$

- special case : rotation

$$T_{\Theta} \in SO(2)$$

$$(T_{\Theta} f)(\varphi) := f(\varphi - \Theta)$$

$$a'^{\ell} = \langle (T_{\Theta} f), \frac{1}{\sqrt{2\pi}} Y^{\ell} \rangle = \frac{1}{\sqrt{2\pi}} \int (T_{\Theta} f)(\varphi) \overline{Y^{\ell}(\varphi)} \, d\varphi$$

$$= \frac{1}{\sqrt{2\pi}} \int f(\varphi - \Theta) \overline{Y^{\ell}(\varphi)} \, d\varphi$$

$$\overline{(X + iY)} = (X - iY)$$

$$= \frac{1}{\sqrt{2\pi}} \int f(\varphi) \overline{Y^{\ell}(\varphi + \Theta)} \, d\varphi$$

$$= \frac{1}{\sqrt{2\pi}} \overline{Y^{\ell}(\Theta)} \int f(\varphi) \overline{Y^{\ell}(\varphi)} \, d\varphi$$

$$= \overline{Y^{\ell}(\Theta)} \langle f, \frac{1}{\sqrt{2\pi}} Y^{\ell} \rangle$$

$$= \overline{Y^{\ell}(\Theta)} \cdot a^{\ell}$$

$$(T_{\Theta} a^{\ell}) := \underbrace{\overline{Y^{\ell}(\Theta)}}_{\text{image representation}} . \underbrace{a^{\ell}}_{\text{coefficient}}$$

a^{ℓ}: complex, irreducible